The Relational Model 1

- Tables vs. relations
- Relation schema
  - –Definition in SQL
- Integrity Constraints – Primary Key
  - –Definition in SQL
- Integrity Constraints – Foreign Key
  - –Defintion using SQL
- Summary

Readings: Sections 2.1—2.3; 2.5 of the textbook.

The Village Cinema Database

http://www.villagecinemas.com.au/

- Facts to keep in the database:
  - Movie: mvID, title, rating, release date, length (in minutes), producer studio. mvID is an artificial attribute (not natural information).
  - Classification: a movie may be associated with several genres (comedy, action, violence, adventure, drama).
  - Cast: A movie has many actors, and an actor can perform in many movies.
  - Direct: A movie can have several directors (usually less than three) and a director can direct many movies.

A relational database is a set of tables

The Village Cinema database

Movie

mvID	Title	Rating	Rel_date	Length	Studio
1	Angels & Demons	M	14-05-2009	138	Sony Pictures
2	Coco Avant Chanel	PG	25-06-2009	108	Roadshow
3	Harry Potter and the Half-Blood Prince	M	15-07-2009	153	Roadshow
4	The Proposal	PG	18-06-2009	107	Disney
5	Ice Age: Dawn of the Dinosaurs	PG	01-07-2009	94	20th Century Fox

Classification

mvID	Genre
1	Drama
2	Drama
3	Drama
3	Action
3	Adventure
4	Comedy
5	Comedy
5	Animated

Cast

mvID	Actor
1	Tom Hanks
2	Audrey Tautou
2	Benoît Poelvoorde
2	Alessandro Nivola
2	Marie Gillain
3	Daniel Radcliffe
3	Emma Watson
3	Rupert Grint
4	Sandra Bullock
4	Ryan Reynolds
4	Malin Akerman
4	Mary Steenburgen
4	Betty White
5	John Leguizamo
5	Queen Latifah
5	Denis Leary
5	Ray Ramono
5	Chris Wedge

Direct

mvID	Director
1	Ron Howard
2	Anne Fontaine
3	David Yates
4	Anne Fletcher
5	Carlos Saldanha
5	Mike Thurmeier

A relational database is a collection of tables …

What does a table comprise?

- Table name: Movie.
- Table header defines columns of the table
  - *mvID, Title, Rating, Rel_date, Length, Studio*
- Rows of data
  - 5 movies.

Movie

mvID	Title	Rating	Rel_date	Length	Studio
1	Angels & Demons	M	14-05-2009	138	Sony Pictures
2	Coco Avant Chanel	PG	25-06-2009	108	Roadshow
3	Harry Potter and the Half-Blood Prince	M	15-07-2009	153	Roadshow
4	The Proposal	PG	18-06-2009	107	Disney
5	Ice Age: Dawn of the Dinosaurs	PG	01-07-2009	94	20th Century Fox

A relational database is a set of relations

- A relation = A table

Relation	Table
relation schema	table header
attribute	column
tuple	row
tuple component	cell

Relation Schemas

- The name of a relation and the set of attributes for the relation is called the *schema* for the relation. For example,

  Movie(mvID, Title, Rating, Rel_date, Length, Studio).

- Each attribute has a *domain* --- the set of all possible values allowed for the attribute.
- The rows of a relation are *tuples* comprising *components* that are *atomic* --- a component can not be broken into smaller components.
- Each component of a tuple is a value from the corresponding attribute domain.

Relation Schemas …

- The schema of the Movie relation is
  Movie(mvID, Title, Rating, Rel_date, Length, Studio).
- The domain of mvID is Integer (data type).
- The current instance of Movie relation has 5 tuples. The first tuple of Movie relation has 6 components:
  - *1, 'Angels & Demons', M, 14-05-2009, 138, 'Sony Pictures'.*

  Note that the date '14-05-2009' is an atomic value.
- The current instance of the Village Cinemas database comprises 4 relations.

Relational Terminology

- *Relation Instance* = relation schema + data
- *Database schema* = set of all relation schemas in the database.
- *Database* = a set of relation instances.

The term *Relation* can refer to Relation schema and Relation Instance. The term *Database* can refer to Database schema and Database Instance. The meaning of these terms can usually be distinguished from the context.

Database Schemas in SQL

- SQL is a data definition language for defining database schemas and constraints.
  - The CREATE TABLE statement defines and creates an empty table.

The CREATE TABLE statement

```
CREATE TABLE Movie(
    mvID     INTEGER,
    title    VARCHAR(40),
    rating   CHAR(2),
    rel_date DATE,
    length   INTEGER,
    studio   VARCHAR(20)
);
```

mvID	Title	Rating	Rel_date	Length	Studio

Elements of Table Definition

- Most basic element: an attribute and its type.
- The most common types are:
  - INT or INTEGER (synonyms).
  - REAL or FLOAT (synonyms).
  - CHAR(*n*) = fixed-length string of *n* characters.
  - VARCHAR(*n*) = variable-length string of up to *n* characters.

SQL Values

- Strings require single quotes.
  - E.g., ‘Tom Hanks’, and ‘The Proposal’.
  - Two single quotes = real quote, e.g., ’Mary”s little lamb’.
- Any value can be NULL.
  - NULL can have several semantics:
    - Value unknown, inapplicable, or withheld.

SQL Values …

- DATE and TIME are types in SQL and so there are DATE and TIME values.
- The default form of a date value is 'DD-MM-YYYY'.
- The form of a time value is: 'hh:mm:ss', with an optional decimal point and fractions of a second following.
  - `TIME '15:30:02.5'` = two and a half seconds after 3:30PM.

Integrity constraints – Key and Primary Key

- An attribute or list of attributes are a *Key* of a relation if no two tuples of the relation may agree in all the attribute(s) on the list. If there are several candidate keys, one is specified as the *Primary Key*.
  - No two movies can have the same value for *mvID*. *mvID* is the artificial key of Movie.
  - A movie (represented by *mvID*) can be linked to multiple genres, and a genre can link to multiple movies. So the *mvID* and genre together – {*mvID, Genre*} -- are the only key for Classification.
- Primary key attributes are underlined.
  - *Movie(mvID, Title, Rating, Rel_date, Length, Studio)*
  - *Classification(mvID, genre)*

Key and Primary Key …

- Note that the key constraint is a natural property of real-world data and part of a relation schema. Key definition can only be derived from the real-world. Although sample data in a relation can help disapprove a possible key definition, they can not be used to prove a key definition.
- Example: For the Direct relation, when only the director information for the first 4 movies are kept in the relation. Even though the data may suggest that each movie only has one director, and mvID is the primary key. This is not a property that is always true.

Direct

mvID	Director
1	Ron Howard
2	Anne Fontaine
3	David Yates
4	Anne Fletcher

Defining Primary Keys in SQL

- A primary key definition PRIMARY KEY (…)
  is another element in the list of elements of a CREATE TABLE statement.
  - Movie(mvID, title, rating, length, studio)

```
CREATE TABLE Movie(
    mvID     INTEGER,
    title    VARCHAR(40),
    rating   CHAR(2),
    rel_date DATE,
    length   INTEGER,
    studio   VARCHAR(20),
    PRIMARY KEY (mvID)
);
```

Defining Primary keys …

- Several attributes together make up a key.
  - Classification (<u>mvID, Genre</u>)

Classification

mvID	Genre
1	Drama
2	Drama
3	Drama
3	Action
3	Adventure
4	Comedy
5	Comedy
5	Animated

```
CREATE TABLE Classification(
    mvID     INTEGER,
    genre    CHAR(10),
    PRIMARY KEY (mvID, genre)
);
```

Ensuring Data Integrity –
Tuples can not have the same value for all PK attributes.

- With the given Movie relation schema and data

  Create table Movie (.. Primary key (mvID));

  Movie(mvID, Title, Rating, Rel_date, Length, Studio)

mvID	Title	Rating	Rel_date	Length	Studio
1	Angels & Demons	M	14-05-2009	138	Sony Pictures
2	Coco Avant Chanel	PG	25-06-2009	108	Roadshow
3	Harry Potter and the Half-Blood Prince	M	15-07-2009	153	Roadshow
4	The Proposal	PG	18-06-2009	107	Disney
5	Ice Age: Dawn of the Dinosaurs	PG	01-07-2009	94	20th Century Fox

- Insert a new movie record to the relation (SQL syntax will come later)

```
SQL> insert into movie values (3, 'Transformers: Revenge of the Fallen', 'M', '24-06-2009', 150, 'Paramount');
                *
ERROR at line 1:
ORA-00001: unique constraint (ZHANG.SYS_C00112890) violated
```

Error here!

Ensuring Data Integrity:
-- Null is not allowed for PK attributes.

SQL> insert into movie values(null, 'My Life in Ruins', 'PG', '16-07-2009', 95, '20th Century Fox')

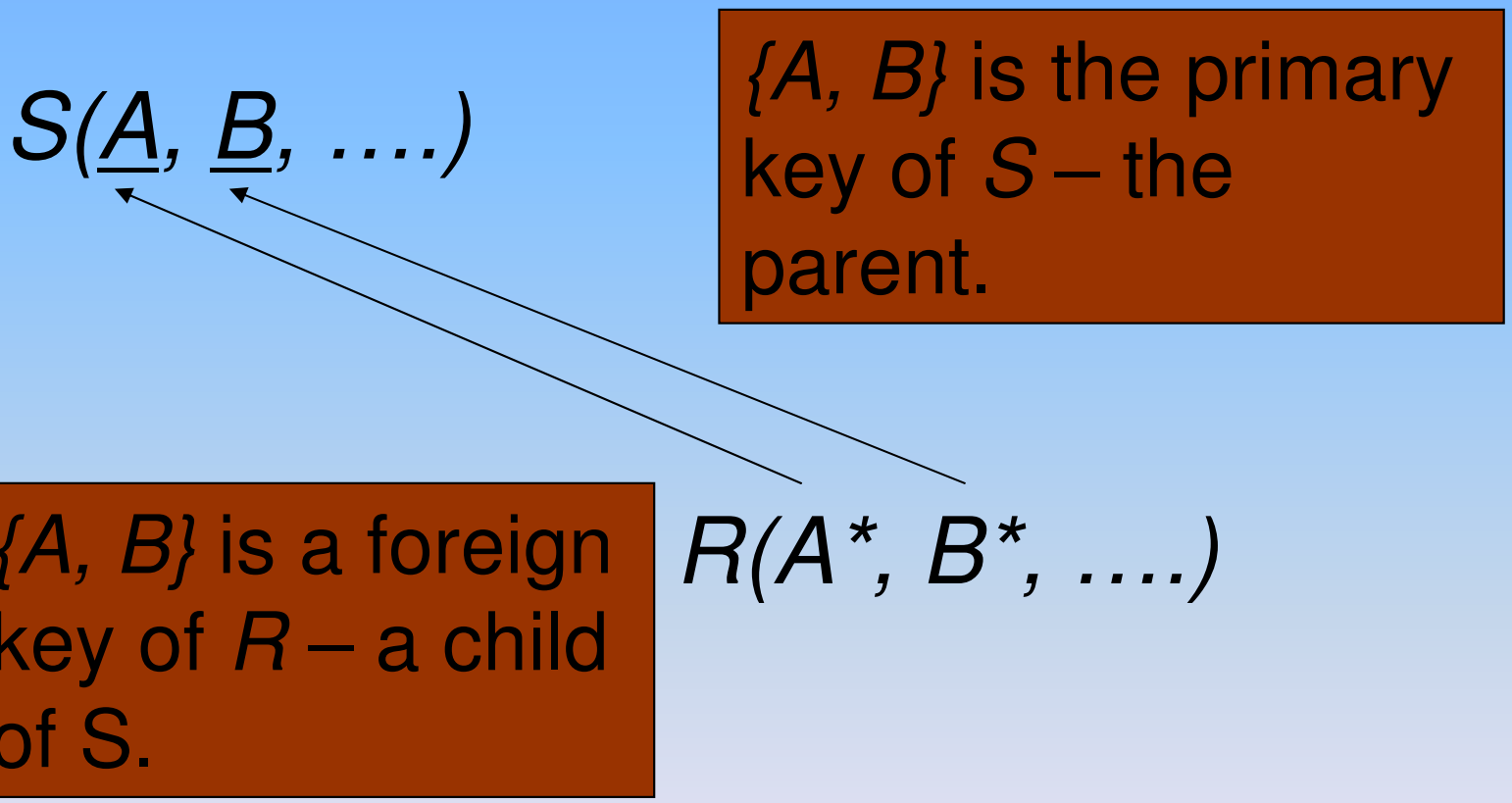
← Error here!

ERROR at line 1:
ORA-01400: cannot insert NULL into ("ZHANG"."MOVIE"."MVID")

Integrity Constraints -- Foreign Keys

- Values appearing in attributes of relation *R* must appear in the primary key attributes of another relation *S*. These attributes comprise a foreign key in *R* that references the primary key of S.
- Foreign keys in a relation schema are usually denoted by an asterisk (*).

Foreign Keys …



The Foreign-Key Constraint

- If there is a foreign-key constraint from relation *R* to relation *S*, the constraint *prevents* introducing insensible data:
  - An insert or update to *R* introduces values not found in *S*.
  - A deletion or update to S causes some tuples of *R* to "dangle."

The Village Cinema DB – Primary and Foreign Keys

Movie(mvID, Title, Rating, Rel_date, Length, Studio)

Classification(mvID*, Genre)

Cast(mvID*, Actor)

Direct(mvID*, Director)

Defining Foreign Keys

- A foreign key definition

  FOREIGN KEY (….) REFERENCES *relation(attributes)*

  can be added as an element in the CREATE TABLE statement.

Defining Foreign Keys …

```
CREATE TABLE Classification(
   mvID     INTEGER,
   genre    CHAR(10),
   FOREIGN KEY (mvID) REFERENCES Movie(mvID)
);

CREATE TABLE Cast(
   mvID     INTEGER,
   actor    VARCHAR(20),
   FOREIGN KEY (mvID) REFERENCES Movie(mvID)
);

CREATE TABLE Direct(
   mvID     INTEGER,
   director VARCHAR(20),
   FOREIGN KEY (mvID) REFERENCES Movie(mvID)
);
```

The Village Cinema DB – Primary and Foreign Keys in SQL

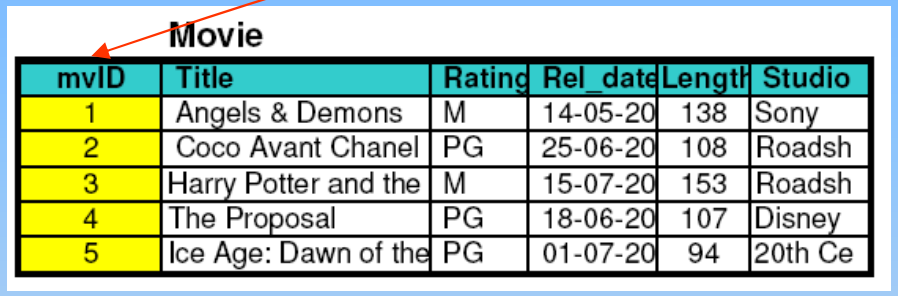
```
CREATE TABLE Movie(
   mvID     INTEGER,
   title    VARCHAR(40),
   rating   CHAR(2),
   rel_date DATE,
   length   INTEGER,
   studio   VARCHAR(20),
   PRIMARY KEY (mvID)
);

CREATE TABLE Classification(
   mvID     INTEGER,
   genre    CHAR(10),
   PRIMARY KEY (mvID, genre),
   FOREIGN KEY (mvID) REFERENCES Movie(mvID)
);

CREATE TABLE Cast(
   mvID     INTEGER,
   actor    VARCHAR(20),
   PRIMARY KEY (mvID, actor),
   FOREIGN KEY (mvID) REFERENCES Movie(mvID)
);

CREATE TABLE Direct(
   mvID     INTEGER,
   director VARCHAR(20),
   PRIMARY KEY (mvID, director),
   FOREIGN KEY (mvID) REFERENCES Movie(mvID)
);
```

The Village Cinema DB – Primary and Foreign Keys …

- A movieID that does not appear in the Movie relation can not be added into the Classification, Cast or Direct relations.
- A tuple (describing a movie) can only be deleted from the Movie relation if there its mvID does not appear in Classification, Cast or Direct. In other words, dangling mvID values in Classifiction, Cast or Direct are not allowed.

Ensuring Data Integrity:
-- Foreign keys ensure sensible Insertion:

Create table Direct(... foreign key (mvID) references Movie(mvID));



Movie

mvID	Title	Rating	Rel_date	Length	Studio
1	Angels & Demons	M	14-05-20	138	Sony
2	Coco Avant Chanel	PG	25-06-20	108	Roadsh
3	Harry Potter and the	M	15-07-20	153	Roadsh
4	The Proposal	PG	18-06-20	107	Disney
5	Ice Age: Dawn of the	PG	01-07-20	94	20th Ce

Direct

mvID	Director
1	Ron Howard
2	Anne Fontaine
3	David Yates
4	Anne Fletcher
5	Carlos Saldanha
5	Mike Thurmeier

```
SQL> insert into direct values(7, 'The Wiggles')
          *
ERROR at line 1:
ORA-02291: integrity constraint (ZHANG.SYS_C00112910) violated
  - parent key not found
```

Ensuring Data Integrity:
-- Foreign keys ensure sensible Deletion:

- A movie tuple can only be deleted when it is not referenced by any other relations.

```
/* To delete the movie "Happy Potter" from the database. */
SQL> delete from movie where mvID = 3;
delete from movie where mvID = 3
*
ERROR at line 1:
ORA-02292: integrity constraint (ZHANG.SYS_C00112906) violated - child record found


/* To remove the movie relation. */
SQL> drop table movie;
drop table movie
           *
ERROR at line 1:
ORA-02449: unique/primary keys in table referenced by foreign keys
```

Other Matters

- In Oracle, case of letters is not distinguished in identifiers and key words:
  - Insert = INSERT = InSerT
  - Movie = movie = movIE
  - But ‘Paramount’ ≠ ‘paramount’ ≠ ‘PARAMOUNT’
- “;” or “/” (but not both!) terminates a query and get it executed;

Other Matters ...

- Write/Edit a query using a text editor and save in a file
  - Unix: Vi, pico, xedit, emacs
  - Windows: Notepad
- Executing a file containing SQL commands.

  SQL> @filename.sql
- Editing a query inside SQL*Plus using “vi”.

  SQL> edit

Summary

- Reading: Sections 2.1—2.3; 2.5 of the textbook.
- Tables and relations are used interchangeably, even though relations have a much stricter mathematical definition.
- Primary key and foreign key are the most important *integrity constraints* in the relational model. They help make sure that the data kept in databases are sensible.
  - Other integrity constraints can be defined in Oracle.
- SQL as a *data definition language*.
  - CREATE TABLE ... (

```
    ....
  PRIMARY KEY (...)
   FOREIGN KEY (...) REFERENCES ....
);
```